Technical Report: BioDCASE 2026

Baleen Whale Deep Learning Detection and Classification Network

Contributors:

- Liz Ferguson[1]
- Gabriela Alongi[1]

[1] Ocean Science Analytics LLC, San Diego, California, 92129, USA

Tool Introduction

DeepAcoustics is a deep learning-based tool designed for the detection and classification of underwater acoustic signals, with a focus on marine mammal vocalizations. Developed with a modular architecture, DeepAcoustics allows users to configure, train, and apply custom neural network models for sound event detection. DeepAcoustics supports a range of deep learning architectures for acoustic detection and classification, including Tiny YOLO, ResNet, and Darknet-based networks, allowing users to select the model best suited for their specific data and computational needs.

This graphical user interface (GUI)-based tool was implemented in MATLAB (version R2024b, developed by MathWorks), leveraging the following MATLAB toolboxes: Computer Vision, Curve Fitting, Deep Learning, Image Processing, Parallel Computing, Signal Processing, and Statistics and Machine Learning. Over the past several years, DeepAcoustics has undergone continuous development and recently incorporated multiclass detection capabilities, all within a user-friendly framework that supports efficient spectrogram preprocessing, model training, testing, and deployment. The modifications to the multi-class components have been extensive this past year after our trial with the BioDCASE 2025 challenge. For more information on DeepAcoustics and the network training process, please refer to Sugarman et al., 2025.

Key Observations & Issues

We had several interesting observations from this year's challenge that are worth noting.

1. In our recent efforts with a multiclass network training for a Pacific low frequency baleen whale network, we found we found that data with a 400 Hz sample rate provided better spectral and temporal window sizes for training samples. We had also tested out data that was 250 Hz sample rate, and network training was suboptimal. So we attempted several different image resolutions to try and mitigate the issue, and found that the v1 (which we decided to submit) did a better job of detecting more of the calls, but that detection was

offset sometimes for bmD calls, as compared to the v2 and v3 versions. We also noticed very poor performance for the fin whale calls which we this is related to this. The v2 and v3 versions did much better job of boxing the bmD calls and the fin whale 20 Hz and 20Hz plus calls, but there were fewer calls detected overall. The overlap difference could contribute to this difference but would not explain the full differentiation. Frequency resolution is calculated by dividing the sample rate (in Hz) by the FFT size (also in Hz). The temporal resolution is calculated by dividing the FFT size by the sample rate. Below is a table comparing the image settings for training and validation images:

Network version	Window Size (FFT/NFFT)	Image resolution (pixels)	Overlap %	Optimized Image Length (s)	Frequency Resolution (Hz)	Temporal Resolution (s)
v1	224	320	70	86.02	1.11	0.896
v2	128	320	80	32.77	1.95	0.51
v3	140	320	70	53.76	1.78	0.56

Figure 1 provides an example of the issues with call selection that are resolved with the lower FFT size (version 2 is the green boxes), but sacrifices a larger number of calls being detected. We believe this could be somewhat mitigated by using a 400 Hz sample rate for training files which would provide a courser frequency resolution of 1.78 Hz, but a better temporal resolution of 0.56 seconds, which is better for shorter duration calls such as the bmD and bp20, 20 plus and 40 Hz calls.

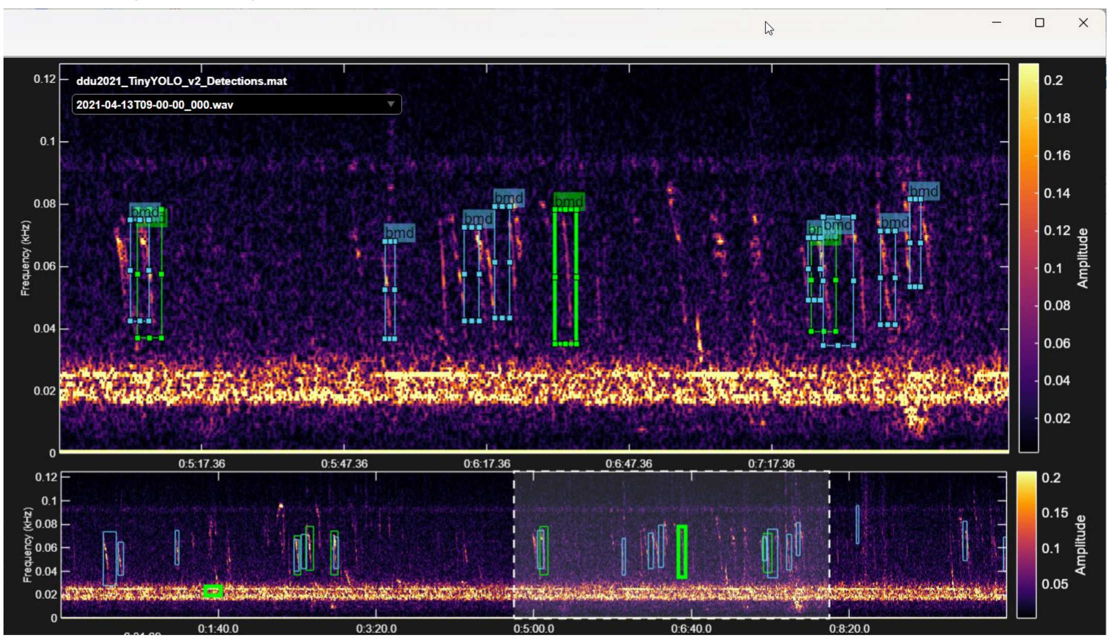

Figure 1: bmD call Detections from TinyYOLO v1 in blue boxes and v2 in the green boxes.

2. We found overfitting happening with the 30 epoch selection for v1 (the submitted network). We ran another network that was only 10 epochs and got the exact same results so did not think the overfitting impacted the results.

We had issues importing the third kerguelen2015 validation dataset and so did not use that in our training.

Network Details

For the baleen whale detection task, DeepAcoustics, again this year proceeded with the Tiny YOLO network, a lightweight convolutional neural network consisting of 74 layers. Tiny YOLO offers a compact alternative to full-scale YOLO architectures, optimized for real-time detection without sacrificing significant accuracy. This network was selected due to its efficiency and proven performance in constrained computational environments.

The Tiny YOLO model was pre-trained on the COCO dataset, a comprehensive object detection dataset containing over 330,000 labeled images. Unlike datasets such as ImageNet that focus on bounding boxes, COCO provides object segmentation, which enhances the model's ability to recognize fine-grained structures in data. This pre-training served as the foundation for transfer learning, enabling the network to adapt effectively to spectrogram image inputs of baleen whale calls.

Training and validation images were generated from audio files the settings provided below. The selected image length (which the user can define) was set to “optimize” wherein the temporal resolution is retained given the selected window size (i.e., FFT/NFFT). We have found in previous efforts that by increasing the image length to the full duration possible without losing temporal resolution, the tool is able to better learn the noise in the dataset (in addition to the randomly generated noise that is produced and serves as a separate category). We are not bound to the same image length as binary classification systems because DeepAcoustics is able to use multiple bounding box annotations within a given image for training, focusing on the bounding box limits as opposed to just defining anything within a set duration box. The TinyYOLO model architecture and training parameters were selected based on recent optimal efforts for a similar set of data from the Pacific. They include:

Parameter	Value
Image resolution	320 x 320 pixels
Image duration	86.02 seconds
Window size & NFFT	224
Overlap	70%
Anchors	9
Optimization algorithm	RMSprop
Learning rate	0.0005
Mini-batch size	64
Epochs	30
Training duration	23 hr: 39 min

Sugarman, P. C., Ferguson, E. L., Alongi, G. C., Schallert, J. P., & Lyn, H. (2025). Effects of network selection and acoustic environment on bounding-box object detection of delphinid whistles using a deep learning tool. The Journal of the Acoustical Society of America, 157(6), 4613-4627.

Detection Output

The trained DeepAcoustics network was evaluated on all four evaluation datasets and compiled in the results csv. The model was evaluated on the validation data using the Performance Metrics tool in DeepAcoustics to get a more quantitative sense of performance, although since these were used to train the network, the results are likely biased. Detection performance metrics were computed at an intersection-over-union (IOU) threshold of 0.1, reflecting the inherent variability in manual annotation of animal vocalizations where bounding box boundaries are subjective and call detection rather than precise localization is the primary objective. Even so, overall performance was uneven across call types and datasets (F1: 0.332, Precision: 0.659, Recall: 0.222), with performance varying by dataset: Casey 2017 (F1: 0.482, Precision: 0.632, Recall: 0.390) and Kerguelen 2014 (F1: 0.259, Precision: 0.686, Recall: 0.160). Kerguelen 2015 was excluded from the quantitative evaluation due to data import issues.

Enclosed with Submission Packet:

1. This report
2. OSA_TinyYOLO_task2_1.meta.yaml
3. OSA_TinyYOLO_task2_1.output.csv